

# 号码认证接入文档

## 1、环境配置

### 1.1 流程说明

调用 获取授权码（2.1） —成功后—> 调用 置换接口（2.2）

注：onFailed 指未进行网络访问，在逻辑层被打回，而 onSuccess 大部分返回的是进行网络访问后的结果，code 为 1 时返回的 response 为 null

### 1.2 资源拷贝

- aar 拷贝至 libs 目录

### 1.3 混淆配置

项目开启混淆时避免混淆 sdk（sdk 带有混淆）

```
-dontwarn com.sdk.**  
  
-keep class com.sdk.** { *;}
```

### 1.4 SDK 初始化环境

```
SDKManager.init(getApplicationContext(),"分配给您的密钥"," 分配给您的公钥" );
```

## 2、接口调用

注：示例默认使用联通能力，如需使用电信能力只需将 **OauthManager** 换为 **OauthManagerCtc** 即可

### 2.1 获取授权码

接口名称：

`OauthManager.getInstance(this).getAuthoriseCode(int timeOut, Callback<T> callBack);`

接口说明：code=0 代表成功，返回的结果只能使用一次，且有效时间为 **30 分钟**

参数说明：

参数	说明
timeOut	超时时间
callBack	回调，详情查看

调用示例：

```
OauthManager.getInstance(this).getAuthoriseCode(10,new Callback<Object>() {  
  
    @Override  
  
    public void onSuccess(int code, String msg, int status, Object response,  
String seq) {  
  
        // do something  
  
    }  
  
    @Override  
  
    public void onFailed(int code, int status, String msg) {
```

```
        // do something

    }

})
```

## 2.2 置换接口

### 2.2.1 sdk 置换

接口名称:

```
OauthManager.getInstance(this).getMobileForCode(String accessCode, String mobile, int
        timeOut,final Callback<T> callBack);
```

**接口说明:** 此接口可达到置换效果，但**安全性不高**（没有经过接入方服务器），所以此方式供 **demo** 演示

参数说明:

参数	说明
accessCode	授权码
mobile	需要认证的号码
timeOut	超时时间
callBack	回调，详情查看

调用示例:

```
OauthManager.getInstance(this).getMobileForCode(accessCode,mobile,10,new
        Callback<Object >() {

            @Override
```

```
        public void onSuccess(int code, String msg, int status, Object response,
String seq) {

            // do something

        }

        @Override

        public void onFailed(int code, int status, String msg) {

            // do something

        }

    })
```

### 2.2.2 服务端置换

[详情查看](#) [服务端接入文档](#)

## 3、常见问题

- 可以提供 jar 包吗？  
答：aar 解压后就是 jar，再将需要的资源拷贝至项目即可。
- sdk 提供的都是混淆的代码，如何具体参数的含义？  
答：[类介绍](#)中包含所有的类描述、字段、方法的介绍

## 4、SDK 状态码（sdk 状态码为 6 位数）

**注：关于 code（只适用于 SDK 状态码）：**

**code=0 代表完全成功**

**code=1 代表各种问题导致的失败，详情查看对应返回的 status，下面是各类**

## 状态码的说明

状态码 (status)	信息 (msg)	示例说明
100000	通用错误	一般发生在内部，会伴有异常日志
100001	网络访问出错	apiKey 配置错误或者 404 错误
100002	参数错误	接口所需的基本参数未填写，详情查看对应的 msg
100003	数据格式错误	服务器返回结果 SDK 解析出错
100004	无结果	服务器响应返回 null
100005	数据不匹配	短信上行的认证结果与填入的不匹配计为认证失败
100006	包名不匹配	服务器返回的结果非本 app 计为失败
100007	apikey 为空	应用未填入 apiKey 密钥
100008	超时	超过自定义的超时时间
100012	无公钥	应用未填入公钥
100016	公钥出错	公钥填写有误，加密时抛出异常
100018	用户取消登录	预取号成功后，用户点击返回
100019	SDK 解密失败	服务器返回数据时，sdk 会预先进行解密
100020	操作频繁请, 稍后再试	10 分钟内只能访问 30 次

## 5、服务端状态码（服务端状态码为 3、4 位数）

状态码 (status)	信息 (msg)
100	成功（有数据返回）
103	成功（无数据返回）

状态码 (status)	信息 (msg)
101	鉴权失败
104	系统繁忙
111	认证成功，（手机号码和认证手机号码不一致，参与计费）
112	认证失败（未取到手机号码）
2001	取号失败/认证失败
2002	授权码为空
5001	序列号不存在
6001	参数信息错误
6002	应用信息错误
6003	sdk 信息错误
6004	SIM 信息错误
6005	接入信息解析错误
6006	获取接入信息流控类型错误
6008	流控值超限
6009	设备信息错误
6010	应用密钥信息不匹配
7001	数据更新失败
8001	无支付权限
8002	余额不足